Specification of Tokens

Lecture 2
Section 3.3

Robb T. Koether

Hampden-Sydney College

Fri, Jan 16, 2015

Outline

Alphabets and Strings

Definition (Alphabet)

An alphabet is a finite set of symbols. Traditionally, we denote an alphabet by the letter Σ.

Definition (String)

A string is a finite sequence of symbols.

Definition (Empty String)

The empty string, denoted ε, is the string that contains no symbols. The empty string has length 0.

Alphabets and Strings

Example (Alphabets and Strings)

- Examples:
  - The traditional alphabet is

$$\Sigma = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \ldots, \mathsf{Z}\}.$$

  - The binary alphabet is $\Sigma = \{0, 1\}$.
  - For C programs, the alphabet is the set of ASCII characters.

Languages

Definition (Language)

A language is a set of (finite) strings over a given alphabet.

- A language can be (and usually is) infinite.
- The set of all even integers over the alphabet $\Sigma = \{0, 1, \ldots, 9\}$ is a language.
- The set of all C programs is a language.

Outline

Operations on Languages

Definition (Language)

Let L, L_1, and L_2 be languages.

- Union:

$$L_1 \cup L_2 = \{x \mid x \in L_1 \text{ or } x \in L_2\}.$$

- Concatenation:

$$L_1 L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}.$$

- Repeated concatenation:

$$L^n = LLL \cdots L \; n \text{ copies of } L.$$

- Kleene closure:

$$L^* = \{\varepsilon\} \cup L \cup L^2 \cup L^3 \cdots .$$

Operations on Languages

Example (Language)

- Let
$$L_1 = \{1, 3, 5, 7, 9\}$$
and
$$L_2 = \{0, 2, 4, 6, 8\}.$$
- Describe $L_1 \cup L_2$.
- Describe $L_1 L_2$.
- Describe $(L_1 \cup L_2)^3$.
- Describe $(L_1 \cup L_2)^* L_2$.

Outline

Regular Expressions

- A regular expression can be used to describe a language.
- Regular expressions may be defined in two parts.
  - The basic part.
  - The recursive part.

Regular Expressions

- The basic part:
  - ε represents the language $\{\varepsilon\}$.
  - $\mathbf{a}$ represents the language $\{a\}$ for every $a \in \Sigma$.
  - Call these languages $L(\varepsilon)$ and $L(\mathbf{a})$, respectively.

Regular Expressions

- The recursive part: Let r and s denote regular expressions.
  - $r \mid s$ represents the language $L(r) \cup L(s)$.
  - rs represents the language $L(r)L(s)$.
  - r^* represents the language $L(r)^*$.
  - (r) represents the language $L(r)$.

- In other words
  - $L(r \mid s) = L(r) \cup L(s)$.
  - $L(rs) = L(r)L(s)$.
  - $L(r^*) = L(r)^*$.
  - $L((r)) = L(r)$.

Example

Example (Identifiers)

- Identifiers in C++ can be represented by a regular expression.

$$r = \mathtt{A} \mid \mathtt{B} \mid \cdots \mid \mathtt{Z} \mid \mathtt{a} \mid \mathtt{b} \mid \cdots \mid \mathtt{z}$$
$$s = \mathtt{0} \mid \mathtt{1} \mid \cdots \mid \mathtt{9}$$
$$t = r(r \mid s)^*$$

Regular Expressions

Definition (Regular definition)

A regular definition of a regular expression is a "grammar" of the form

$$\begin{aligned} d_1 &\rightarrow r_1 \\ d_2 &\rightarrow r_2 \\ &\vdots \\ d_n &\rightarrow r_n \end{aligned}$$

where each r_i is a regular expression over $\Sigma \cup \{d_1, d_2, \ldots, d_{i-1}\}$.

Example

Example (Identifiers)

- We may now describe C++ identifiers as follows.

$$
\begin{aligned}
\textit{letter} &\rightarrow \mathtt{A} \mid \mathtt{B} \mid \cdots \mid \mathtt{Z} \mid \mathtt{a} \mid \mathtt{b} \mid \cdots \mid \mathtt{z} \\
\textit{digit} &\rightarrow \mathtt{0} \mid \mathtt{1} \mid \cdots \mid \mathtt{9} \\
\textit{id} &\rightarrow \textit{letter}\ (\textit{letter} \mid \textit{digit})^*
\end{aligned}
$$

Regular Expressions

- Note that this definition does not allow recursively defined tokens.
- In other words, d_i cannot be defined in terms of d_i, not even indirectly.

Outline

Extensions of Regular Languages

Definition

We add the following symbols to our regular expressions.

- One or more instances: $r^+ = r\, r^*$.
- Zero or one instance: $r? = r \mid \varepsilon$.
- Character class: $[a_1 a_2 \cdots a_n] = a_1 \mid a_2 \mid \cdots \mid a_n$.

Extensions of Regular Languages

Example (Identifiers)

- Identifiers can be described as

$$\begin{aligned} \textit{letter} &\rightarrow [\texttt{A-Za-z}] \\ \textit{digit} &\rightarrow [\texttt{0-9}] \\ \textit{id} &\rightarrow \textit{letter}\ (\textit{letter} \mid \textit{digit})^* \end{aligned}$$

Extensions of Regular Languages

Example (Floating-point Numbers)

- Floating-point numbers can be described as

$$
\begin{aligned}
\textit{digit} &\rightarrow [\texttt{0}-\texttt{9}] \\
\textit{digits} &\rightarrow \textit{digit}^{+} \\
\textit{number} &\rightarrow \textit{digits}\ (.\ \textit{digits})?\ (\texttt{E}\ [+-]?\ \textit{digits})?
\end{aligned}
$$

Outline

Assignment

Assignment

- Read Section 3.3.
- Exercises 2, 3.